of the International Joint Conference on Artificial Intelligence, Detroit, MI. 1480–1485.

Satta, G. and Stock, O. 1994. Bi-Directional Context-Free Grammar Parsing for Natural Language Processing. To appear in: Artificial Intelligence Journal, Elsevier Science Publishers (North-Holland).

Schwartz, R. and Chow, Y. L. 1990. The N–Best algorithm: an efficient and exact procedure for finding the N most likely sentence hypotheses. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, New Mexico, USA. 81–84.

Soong, F. K. and Huang, E. F. 1991. A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada. 705–708.

Stock, O.; Falcone, R.; and Insinnamo, P. 1989. Bidirectional Chart: A Potential Technique for Parsing Spoken Natural Language Sentences. *Computer Speech and Language* 3(3):219–223.

Stock, O. 1994. Natural language in multimodal human–computer interfaces. To appear in: IEEE Expert.

Storer, J. A. 1985. Textual substitution techniques for data compression. In Apostolico, A. and Galil, Z., editors 1985, *Combinatorial Algorithms on Words*, volume NATO ASI Series, F12. Springer-Verlag, Berlin, Germany. 111 – 129.

Zue, V.; Glass, J.; Goodine, D.; Leung, H.; Phillips, M.; Polifroni, J.; and Seneff, S. 1991. Integration of Speech Recognition and Natural Language Processing in the MIT VOYAGER System. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada. 713–716.

a significant feature, even if this choice disregards the cost of the extra bookkeeping involved in the managing of the grammar generating the N-Best. For the sake of generality of results, which should be as independent as possible both from the acoustic module and the domain application, all the N hypotheses given by the acoustic recognizer have been considered, without stopping at the first linguistically acceptable one. In fact, probably because the subdomain on which the experiments were made is very restricted and the sentences are quite short (7 words per sentence on the average), the correct hypothesis is very often in a very high position among the N hypotheses (we have chosen $N = 10$).

We found out that when the compression algorithm is used, the edges introduced into the chart are about 1/4 than in the sequential case: the average of the ratio between the edges in the two cases is 0.249 (0.248 if the best and the worst cases are eliminated), with a standard deviation of 0.033 (0.022 without the best and the worst case).

Conclusions and future works

A compact representation of the N-Best output has been presented, which well couples with a bidirectional linguistic processing. Such representation is based on a non-recursive context-free model, and is therefore more powerful than normally used lattice representations. The compression is accomplished using an algorithm based on the subword tree of the string obtained concatenating the N sequences interleaved by N different delimiters.

Some experiments have been made whose results, although partial, encourage us to follow this direction. One point which can probably improve the compression is the score used in building the grammar.

An important application of this framework will be its use for error-correcting problems.

Acknowledgements

We would like to thank Oliviero Stock for his useful suggestions and many fruitful discussions and Giorgio Satta for reading a previous version of this paper. We would also like to thank an anonymous referee for his note on memoization, another way of looking at dynamic programming.

References

Agnäs, M. S.; Alshawi, H.; Bretan, I.; Carter, D.; Ceder, K.; Collins, M.; Crouch, R.; Digalakis, V.; Ekholm, B.; Gambäck, B.; Kaja, J.; Karlgren, J.; Lyberg, B.; Price, P.; Pulman, S.; Rayner, M.; Samuelsson, C.; and Svensson, T. 1994. Spoken language translator: First–year report. Technical Report SICS R94:03,SRI CRC–043, Swedish Institute of Computer Science, Stockholm, Sweden, SRI International, Cambridge, England.

Angelini, B.; Brugnara, F.; Falavigna, D.; Giuliani, D.; Gretter, R.; and Omologo, M. 1993. A Baseline of a Speaker Independent Continuous Speech Recognizer of Italian. In *Proceedings of the European Conference on Speech Communication and Technology*, Berlin.

Apostolico, A. 1985. The myriad virtues of subword trees. In Apostolico, A. and Galil, Z., editors 1985, *Combinatorial Algorithms on Words*, volume NATO ASI Series, F12. Springer-Verlag, Berlin, Germany. 85–96.

Bates, M.; Bobrow, R.; Fung, P.; Ingria, R.; Kubala, F.; Makhoul, J.; Nguyen, L.; Schwartz, R.; and Stallard, D. 1993. The BBN/HARC spoken language understanding system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, Mn, USA. II 111–114.

Carpenter, B. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge, England.

Chen, M. T. and Seiferas, J. 1985. Efficient and elegant subword tree construction. In Apostolico, A. and Galil, Z., editors 1985, *Combinatorial Algorithms on Words*, volume NATO ASI Series, F12. Springer-Verlag, Berlin, Germany. 97–129.

Corazza, A.; Federico, M.; Gretter, R.; and Lazzari, G. 1993. Design and acquisition of a task-oriented spontaneous-speech data base. In Roberto, V., editor 1993, *Topics in Intelligent Perceptual Systems, Lecture Notes in Artificial Intelligence*. Springer Verlag, Heidelberg.

Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. 1990. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts london, England.

Fraser, N. M. and Gilbert, G. N. 1991. Simulating speech systems. *Computer Speech and Language* 5(1):81–99.

Kay, M. 1980. Algorithm schemata and data structures in syntactic processing. Technical Report CSL-80, Xerox Palo Alto Research Center, Palo Alto, California.

Rodeh, M.; Pratt, V. R.; and Even, S. 1981. Linear algorithm for data compression via string matching. *Journal of the Association for Computing Machinery* 28(1):16–24.

Satta, G. and Stock, O. 1989. Formal properties and implementation of bidirectional charts. In *Proceedings*

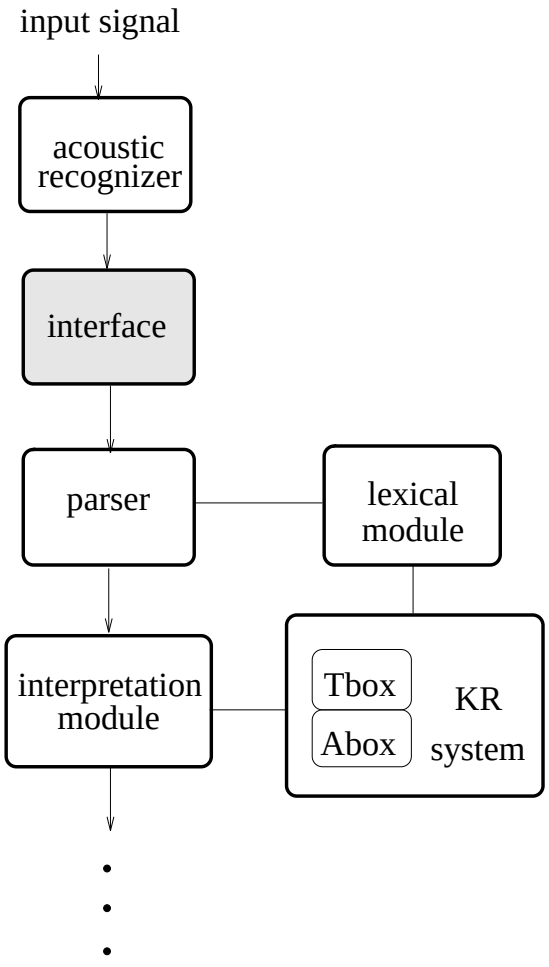

Figure 3: Portions of the architecture relevant for the spoken language system

N-Best candidates are provided, using the algorithm described in [Soong and Huang, 1991]. Briefly, a first pass is performed time synchronously, which computes and stores a forward score for each time and grammar node. Then, an $A^\star$ algorithm is performed backward, which uses as exact predictions the forward scores. Forward and backward scores are combined by means of the tree-trellis mechanism [Soong and Huang, 1991].

The N-Best produced by the acoustic recognizer are then processed by a bidirectional chart-based parser. The features of the parser more relevant to the work presented in this paper have been described in a preceding section. The implementation follows the directions given in [Satta and Stock, 1989], where more details can be found. For the spoken language system, it was decided to encode syntactic and semantic information using a formalism based on Typed Feature Structures (TFS) [Carpenter, 1992].

The parser incrementally interacts with the lexical discrimination module: whenever the parser tries to build a (partially recognized) constituent, the discrimination module is triggered to check the consistency of the semantic part of such a constituent. This check implies the interaction with a knowledge base that models the domain.

Beside checking the semantic consistency of the syntactically correct constituents, the lexical discrimination module produces a kind of logical form which is the first level of semantic representation and constitutes the input for the interpretation module. This module interprets the logical form and retrieves the referents satisfying the user request. Further processing is then accomplished by other modules devoted to the treatment of issues connected with pragmatics and dialog phenomena.

The current implementation can only be used offline. The acoustic component is implemented in the C language while the interface is written in C++ and the linguistic module in Common LISP. Although the online realization is straightforward given the modularity of the architecture, it will be done after some evaluation of the system in terms of understanding results.

Experimental evaluation

The experimental evaluation has been made on a corpus collected as part of the project. This material is used both to train the stochastic language model used by the speech recognition module (i.e. bigrams) and to evaluate the system. For these reasons, the collected material has to be as similar as possible to the input the system is suppose to accept. The Wizard of Oz technique [Fraser and Gilbert, 1991] has been adopted, in which a human "wizard" simulates the system's supposed behavior. So the users interact with a system similar to the one that will be implemented. The collection of the first part of this corpus has been described in [Corazza *et al.*, 1993]. In the work described here, only context independent sentences are used. Moreover, the grammar used by the linguistic module is under development and its actual coverage is quite limited. Therefore, those sentences of the corpus which belong to its intersection with the language generated by the grammar have been chosen.

The goal of the experimental part of the work is the comparison between the proposed approach and the sequential one, i.e. the one in which the linguistic analyzer considers the N-Best one after the other. As an evaluation parameter of the computational effort made by the linguistic analyzer the number of edges introduced during the analysis has been chosen. Given the parser characteristics described above, this seems to be

input is processed locally from left-to-right. Therefore, another algorithm performing a higher compression rate is chosen, even at the cost of a greater asymptotic computational complexity. The algorithm eventually implemented is based on dynamic programming [Cormen *et al.*, 1990] and has quadratic time complexity. A score has been introduced which takes into account the length of the shared chunks in the right-hand side of every rule. The information about shared chunks is given by the substring tree. Therefore, the problem becomes to find the derivation of the input string having optimal score. The grammar producing only such derivation is the expected result. The resulting grammar is in Chomsky normal form.

An example of the output of the algorithm is shown in Figure 2 for the same N-Best considered in Figure 1. For clarity, after every nonterminal rule the part of the input spanned by it is shown. Note that the compression is not optimal (the substring "b c d" is not factorized): this is due to the particular choice of the score function, which can be improved with deeper study and more extensive experimentation.

Modifications to the linguistic module

Given the previously mentioned non-recursivity of the grammar that generates the N best sequences, which implies the possibility of defining a partial order relation over its rules, and that the parser is able to carry on the analyses working bidirectionally, the parsing of the compressed representation of the N best sequences is quite straightforward and requires only limited modifications to the basic mechanism of a standard chart parser. On one side, the ordering among the rules assures that when a rule is being analyzed all the nonterminal symbols appearing in its right-hand side have already been taken into account; on the other hand, the bidirectional strategy of the parser allows full exploitation of the advantages of the factorization accomplished through the compression step, because the parser can build partial analyses even if the portion of the input considered does not include the left-corner of the rule involved.

The algorithm scans the rules of the grammar generating the N-best, runs the standard chart parser on the right-hand side of each of them and stores the results of the analyses (i.e., the part of the chart built by the parser) in a table available for further reuse and processing. Before running the parser on the right-hand side of a rule, the algorithm needs to collect the partial analyses previously computed (and stored in the table) and join these portions of chart in a new chart. While joining such portions, it is necessary to detect the edge combinations resulting from the merging of vertices at the extremes of the partial charts and put the corresponding tasks into the agenda in order to restart the parser.

The presentation of the algorithm assumes a certain familiarity with the basic mechanisms of chart parsing [Kay, 1980].

The main steps of the proposed algorithm are the following[2]:

- for each terminal symbol of the grammar, run the standard parser on it and save the result of the computation in the table *computed-partial-analyses*;
- for each rule of the grammar generating the N-best:
  - initialize the chart, i.e. scan the symbols in the right-hand side of the rule collecting the previously computed partial analyses corresponding to each symbol and joining together such partial analyses in a new chart; while doing that it is necessary to check if the merging of vertices at the extremes of the partial analyses can trigger an edge combination and if so put such a combination task into the agenda;
  - restart the chart parser popping elements from the agenda until it does not get empty;
  - save the result of the computation in the table *computed-partial-analyses*.

As is evident from the description of the algorithm, one of the main advantages of the proposed algorithm for the parsing of the N-Best is in its modularity and in the fact that allows the reuse of standard modules.

The Concierge system

As already mentioned, spoken language for the time being is conceived only as a limited subsystem that works on a subset of the domain of the Concierge system. Figure 3 shows a sketch of the portions of the architecture relevant for the spoken language system. The shaded box refers to the interface presented in this paper. The other modules are briefly described in the following.

The speech recognizer is based on Hidden Markov Models (HMMs), and is described in [Angelini *et al.*, 1993]. 37 acoustic-phonetic phone units, modeled with continuous density HMMs, are concatenated to form legal words, which in turn are embedded in a bigram grammar, whose probabilities are estimated on a training set.

[2]For simplicity, the presentation will assume a rather fixed and rigid sequential behavior useful for producing all the parsing trees associated with a given N-Best set. It is possible to implement the algorithm more flexibly in order to adapt it to the more realistic task of obtaining just the first linguistically consistent sentence hypothesis.

both terminal and nonterminal symbols), as depicted in Figure 2 for the same set of sentences considered in Figure 1.

In such grammar the rewriting rules can be rewritten in such a way that the parser can consider them sequentially, analyzing the right-hand side of the first one (that is only composed of terminal symbols), and then storing the results of this analysis and using them, when necessary, for the analysis of the other rules. This is possible only if the grammar does not permit recursion; in fact, it must be possible to order the rules of the grammar in such a way that, given the i-th rule: $X_i \rightarrow Y_i \ldots Y_m$, for each $1 \leq k \leq m$, Y_k belongs to the set of terminal symbols, or $Y_k = X_j$, $j < i$. The rules should be ordered in such a way that the analysis of each hypothesis is completed as soon as all the chunks involved have been analyzed. With such an order, even if the analysis is stopped at the first linguistically acceptable hypothesis, it is guaranteed that no useless analysis is done. Note that with this technique also in case a substring is repeated within the same sentence hypothesis, its analysis is made just once.

This approach is also expected to be useful in an error correcting perspective. Sentence fragments that appear in more than one sentence are more reliable than the others. In this way, the analysis can be started from the most reliable parts, and, when a failure occurs, only the last part of the analysis can be reconsidered, introducing some recovery strategy, as for example, the possibility of substituting a word with a very similar one.

N-Best representation as a compression problem

It is now necessary to define a parameter that, representing a measure of the adaptation to the linguistic module characteristics, has to be optimized. Our goal is to minimize the computational effort made by the parser. For simplicity, the effort made between two successive shift operations is considered constant, even if in this way the different ability that words have to guide the analysis is disregarded. Under this hypothesis, the parameter to be minimized is the number of shifts done by the parser. The number of shifts that the parser must do is given by the total number of occurrences of terminal symbols in the right-hand side of the rewriting rules of the grammar. In conclusion, our goal is to build a non-recursive context-free grammar producing the N best sequences and having the minimum possible number of occurrences of terminal symbols in the rewriting rules.

This problem can also be seen as a compression problem. Literature exists on the so called *textual substitution techniques* for data compression (see for example [Storer, 1985]) in which the problem of compressing texts is solved by the substitution of substrings by pointers to some other repetition of the same substring in some other place of the source text. In the considered N-Best representation problem, the pointers are represented by the nonterminal symbols of the grammar, and their target by the right-hand side of the rule rewriting it. Note that, without loss of generality, the hypothesis can be made that the grammar admits only one rule rewriting each nonterminal symbol: the number of terminal symbols is to be minimized, and the number of nonterminal symbols is not considered relevant and can be as large as necessary. In this case, attention can be restricted to the *off-line* textual substitution algorithms, i.e. to those for which the hypothesis can be made that the entire input string can be read before any of the output is produced. Following the classification given in [Storer, 1985], an *external macro scheme* is used, in which it is possible to distinguish two parts in the compressed string: a *dictionary*, that in our case is represented by the rules guiding the parser strategy, and a *skeleton*, corresponding to the N best sequences. Always in the framework proposed in [Storer, 1985], *overlapping* among the substrings substituted by pointers should be avoided, because it would contradict the context-freeness of the grammar. Moreover the *topological recursion* should be imposed, which interdicts the creation of cycles in the grammar. In conclusion, we want to use what in [Storer, 1985] is called an *EPM (External Pointer Macro)* scheme of compression.

Many algorithms which implement this kind of compression are based on the *subword tree* of the input string. This structure can be built in linear time with respect to the input string length [Apostolico, 1985; Chen and Seiferas, 1985]. In our case the input string is obtained by the concatenation of the N sequences interleaved by N different delimiters (one of them must be put at the end of the string). The use of these delimiters avoids two repetitions of the same substring crossing the boundary between two of the N best sentences.

Initially the algorithm described in [Rodeh *et al.*, 1981] was chosen to make the compression and therefore to construct the representation of the N-Best. This algorithm is linear, but it tends to optimality only in case the input length tends to infinity, which is not at all our case. On the contrary, the fact that in our case the input length is not very big makes the computational difference between a linear and a quadratic algorithm not critical. Both linear complexity and suboptimal compression rate result from the fact that the

chunks are shared by different hypotheses (or by different parts of the same hypothesis). This approach does not require any further computational effort than the acoustic and linguistic analysis. To avoid duplication of the analysis of shared parts of the sentences, a more compact lattice structure is often used. Two building criteria are possible. The lattice can define exactly the language represented by the N best sequences or the factorization can introduce spurious sequences (see Figure 1). This latter choice is justifiable because usually the sentences obtained by piecing fragments of acoustically likely sentences have an acoustic score that, even if not among the N most likely ones, is quite high. In any case, this approach seems undesirable, because it changes the acoustic results in an empirical way, with simplicity of implementation the only justification. If it is decided to represent only the N best sequences, and a left-to-right linguistic analysis strategy is adopted, the lattice can be represented by a tree, which collapses the prefixes of the sentences. This representation can be optimal when using a left-to-right strategy. As shown above, such a constraint need not be imposed, as a more general bidirectional approach can be used.

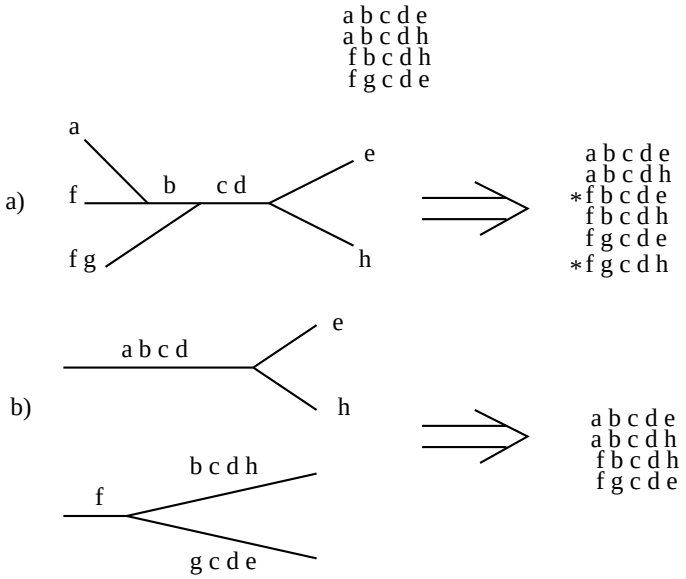

Figure 1: Example of lattice factorization a) with overgeneration; b) without overgeneration.

Therefore, our goal is to find a representation for the N best sequences that couples with a bidirectional parsing strategy and that does not introduce any further hypotheses besides the ones produced by the acoustic front-end. A lattice, being equivalent to a regular representation, is not powerful enough for our purposes. Therefore, a context-free representation of the language generated by the N-Best is used. That is, our goal is to build a context-free grammar generating exactly the language formed by the N best sequences given by the acoustic module and able to guide the parsing strategy in an efficient way.[1] Intuitively, the idea is to substitute common substrings by nonterminal symbols and in correspondence of any such operation to introduce a rule rewriting the nonterminal by the substring (which, in general, can be composed by

output Page 2

```
Input N-Best:

               a b c d e ;
               a b c d h ;
               f b c d h ;
               f g c d e .

Grammar produced by the compression algorithm  (comments are added
for clarity; # indicates the markers -- which in fact are different
one from the other -- between two consecutive hypotheses):

(1,1) => a ;
(2,1) => b ;
(3,1) => c ;
(4,1) => d ;
(5,1) => e ;
(4,2) => (4,1) (5,1) ;  // d e
(3,3) => (3,1) (4,2) ;  // c d e
(6,1) => # ;
(3,4) => (3,3) (6,1) ;  // c d e #
(2,5) => (2,1) (3,4) ;  // b c d e #
// The first hypothesis: a b c d e #:
(1,6) => (1,1) (2,5) ;
(3,2) => (3,1) (4,1) ;  // c d
(2,3) => (2,1) (3,2) ;  // b c d
(11,1) => h ;
(12,1) => # ;
(11,2) => (11,1) (12,1) ; // h #
(8,5) => (2,3) (11,2) ; // b c d h #
// The second hypothesis: a b c d h #
(7,6) => (1,1) (8,5) ;  // a b c d h #
// The following production simply join the two first hypotheses
(1,12) => (1,6) (7,6) ;
(13,1) => f ;
(13,2) => (13,1) (2,1) ; // f b
(13,4) => (13,2) (3,2) ; // f b c d
(18,1) => # ;
(17,2) => (11,1) (18,1) ; // h #
// The third hypothesis: f b c d h #
(13,6) => (13,4) (17,2) ;
// The following production join the third hypothesis to the first
// two
(1,18) => (1,12) (13,6) ;
(20,1) => g ;
(24,1) => # ;
(23,2) => (5,1) (24,1) ; // e #
(21,4) => (3,2) (23,2) ; // c d e #
(20,5) => (20,1) (21,4) ; // g c d e #
// The fourth hypothesis: f g c d e #
(19,6) => (13,1) (20,5) ;
// The following production join the last hypothesis to the other
// three
(1,24) => (1,18) (19,6) ;
```

Figure 2: Output of the compression algorithm applied to the sentence hypotheses considered in Figure 1.

[1] Note that there is no relation between the context-free grammar used to represent the N-Best and the grammar used by the linguistic module; moreover, it is not mandatory that the latter one is context-free.

linguistic analysis is bidirectional.

Because of the complexity of the linguistic models, the computational effort required by the linguistic analysis can be very high. Moreover, while for written interactions no more than one input sequence has to be considered, in speech-based systems the hypotheses to be analyzed can be as many as N. This can be a problem if a working system is required, having an acceptable reaction time.

Even a superficial look at the problem suggests that part of this processing is redundant. In fact, one characteristic of the N-Best is that they usually differ only in some words, having large fragments in common. This means a lot of time would be spent repeating partial analyses that have already been done for other possible solutions. Such duplications could be avoided by factorizing common pieces, which would be analyzed only once, saving the result of the analysis for further reuses. If the linguistic analysis is made following a left-to-right strategy, the factorization can consider the common prefixes of the sentences, generating a tree. On the other hand, our problem is different in the crucial point that the linguistic component of the Concierge system uses a bidirectional parsing strategy. Therefore improvement could come from having a factorization algorithm coupled with it.

In the next section the use of the compressed representation of the N-Best in the interface between acoustic and linguistic modules is described , together with the modifications to the linguistic module it requires. The subsequent section describes the system in which it has been introduced and which we used for the experiments. Then, a section is devoted to the experimental evaluation and a last one to some conclusive considerations and further improvements to be introduced.

The interface design

Characteristics of the linguistic module relevant for the interface

This section describes the characteristics of the linguistic module that are more specifically related to the interface between the acoustic and the linguistic module.

The first step of the linguistic processing is conducted by a chart-based parser [Kay, 1980]. This is a standard choice in the natural language processing field; what is different from the standard (and, as argued in the following, particularly relevant for our kind of approach) is that the parser adopts a bidirectional strategy in analyzing the input sentence, instead of the usual left-to-right one. This means that the parser starts the analysis from several elements within the input sentence and proceeds outward from these triggering elements [Satta and Stock, 1989; Stock *et al.*, 1989].

The choice of the triggering elements can be made either statically or dynamically: in the former case, they are linguistically motivated elements, called *heads* (head-driven strategy, as in written language systems); in the latter case, the choice is made according to characteristics of the elements such as a measure of reliability (island-driven strategy, as for spoken language, in which acoustic score or some other kind of likelihood measure is used as reliability estimation). In the island-driven case both the choice of the elements from which to trigger the analysis and the decision of how to carry on partial analyses can be determined dynamically, depending on the input.

The parser implemented can be customized to behave either as head-driven or island-driven. The only limitation is that it is assumed that there must be at least one triggering element within every grammar rule so that there is no need to deal with the issue of top-down predictions in the portions of the input string that do not contain sufficiently reliable elements (see [Satta and Stock, 1994], for a thorough discussion concerning the various aspects of a general bidirectional approach to parsing). Therefore, this first implementation of the parser is completely bottom-up. Future directions will include the extension of the parser to make use of top-down predictions.

The feature of bidirectionality gives great flexibility (at the cost of a certain computational overhead during the processing, as discussed in [Satta and Stock, 1994]). It is hoped that this flexibility will be particularly useful when dealing with error correction. The bidirectional strategy seems to be particularly suitable to speech applications, given the fact that the output of the acoustic component can contain erroneous, missing and/or redundant words. These phenomena can be dealt with either by correcting such errors (error correcting approach), or by disregarding them (robust parsing approach). In both cases, a bidirectional strategy reduces the risk of founding the analysis on a word that can be wrong.

Context-free N-Best representation

As mentioned in the introduction, the acoustic module produces the N best sequences. They are then put in a form appropriate to the linguistic analysis. The representation to be used obviously depends on the linguistic module characteristics. There are three main options. The most direct method is to consider the output sentences of the acoustic front-end one after the other. This implies that some of the linguistic analyses can be repeated, if, as does happen, sentence

An N-Best representation for bidirectional parsing strategies

Anna Corazza, Alberto Lavelli
Istituto per la Ricerca Scientifica e Tecnologica,
I-38050 Povo/Trento, Italy
e-mail: corazza/lavelli@irst.it

Abstract

In speech understanding systems, the interface between acoustic and linguistic modules is often represented by the N best sequences that match the input signal. They compose a set that will be linguistically analyzed in order to find the interpretation of the input. An appropriate representation of the N-Best could make linguistic processing more efficient. Here a representation based on a context-free model is proposed that is obtained by an algorithm inherited by the data compression field. This algorithm is based on the subword tree of the concatenation of the N best sequences. The proposed representation seems particularly appropriate when coupled with a bidirectional parser and some experiments demonstrate that the approach is worth pursuing. Such experiments focus on the comparison between the proposed representation and a sequential processing of the N hypotheses given by the acoustic module. The comparison takes into consideration the efficiency attained in the two cases, in terms of (partial) analyses constructed by the linguistic module. The obtained results are presented and discussed.

Introduction

The Concierge is a natural language processing system developed at IRST which is able to answer written questions about the structure, staff and research activities of the Institute [Stock, 1994]. Having the aim of exploring the use of spoken input, it seemed advisable to study an architecture that would change the existing modules as little as possible. A loose integration between acoustic and linguistic levels permits building a complete working system as soon as possible and with limited effort. This system can then be used to test new ideas about possible improvements of parts of the system. Moreover, it represents a useful baseline to compare performance of new releases of the complete system. Within this effort a relevant development in the linguistic module is the adoption of bidirectional parsing strategies.

Given the requirement of the loose integration between acoustic and linguistic modules, the most direct solution is to choose an N-Best interface [Schwartz and Chow, 1990; Soong and Huang, 1991] between speech and natural language modules. This paradigm has been widely adopted in the speech processing community [Bates *et al.*, 1993; Zue *et al.*, 1991; Agnäs *et al.*, 1994]. Its main characteristic is a first step consisting of an efficient but not very sophisticated search to restrict the space of admissible solutions to the N most likely sentences. After that, a search based on more powerful knowledge sources can be done on this limited set of hypotheses, reordering them on the basis of more precise scores. Even if the algorithm is not theoretically admissible, it has been shown to be very reliable for reasonable values of N.

The linguistic module receives from the acoustic module the N sentences which obtain the best acoustic score, given only a very simple language model, such as the bigram model. Two approaches are possible. If the linguistic level is able to evaluate a reliability score for every input sentence, it can integrate this score with the acoustic one, reorder the N-Best and consider the one with the highest overall score a solution to the problem. In this case, all the N sentences must be linguistically processed. On the other hand, whenever the linguistic module is simply a YES/NO filter, able only to decide if an input sentence is acceptable or not, it considers these candidate solutions one after the other, stopping the analysis at the first acceptable sentence, given all the syntactic and semantic constraints. Here, cases in which all the N hypotheses have to be considered represent the worst case. Presently the linguistic module of the Concierge is of the latter kind, but some kind of linguistic scores will be introduced in the future. In any case, the proposed approach can be applied in both frameworks, only requiring that the